Amendment to the Claims

This listing of claims will replace all prior versions, and listings, of claims in the application:

1. (Currently Amended) A classfile modification method, comprising:

before loading and using a classfile to create an object during runtime;

converting [[a]] said classfile into a collection of objects, said collection of objects including a first object that represents a method information structure found in said classfile, said collection of objections including a second object that inherits a property that said first object has, said second object representing a bytecode instruction of a method;

adding a third object to said collection of objects, said third object inheriting the properties of said first object, said third object representing a bytecode instruction to be executed by said method that, when executed by said method, invokes a pre-existing dispatcher to identify a plug-in module for said method that said method invokes to report and/or record information about said method;

adding a fourth object to said collection of objects that represents a new method information structure for said classfile, said new method information structure containing byte code instructions for a second method that registers, with said dispatcher upon loading of said classfile, an identity of said classfile's class and respective identities of methods of said classfile, said dispatcher and plug-in module being in existence prior to said loading of said classfile; and,

converting said collection of objects including said third object into a modified version of said classfile.

2-4. (Canceled).

5. (Currently Amended) The classfile modification method of claim [[4]]1, further comprising adding a fifth object to said collection of objects that represents a new field information structure for said classfile.

6. (Previously Presented) The classfile modification method of claim 5, wherein said field information structure is to store a numeric identification assigned to said class by said dispatcher.

7. (Canceled).

8. (Previously Presented) The classfile modification method of claim 1, wherein said third object is added to said collection of objects in a position that corresponds to a region of said method's instructions that is executed just after said method's entry point is reached.

9. (Previously Presented) The classfile modification method of claim 8, wherein said classfile is a Java compatible classfile and said adding of said third_object corresponds to the addition of an invokestatic instruction.

10. (Previously Presented) The classfile modification method of claim 8, wherein said classfile is a Java compatible classfile and said adding of said third_object corresponds to the addition of an invokevirtual instruction.

11. (Previously Presented) The classfile modification method of claim 1, wherein said third object is added to said collection of objects at a position that corresponds to

a region of said method's instructions that is executed if an exit point of said method will inevitably be reached.

12. (Previously Presented) The classfile modification method of claim 11, wherein said classfile is a Java compatible classfile and said adding of said third object corresponds to the addition of an invokestatic instruction.

13. (Previously Presented) The classfile modification method of claim 11, wherein said classfile is a Java compatible classfile and said adding of said third object corresponds to the addition of an invokevirtual instruction.

14. (Previously Presented) The classfile modification method of claim 1, wherein said third object is added to said collection of objects in a position that corresponds to a region of said method's instructions that will be executed if an error arises during execution of said method.

15. (Previously Presented) The classfile modification method of claim 14, wherein said classfile is a Java compatible classfile and said adding of said third object corresponds to the addition of an invokestatic instruction.

16. (Previously Presented) The classfile modification method of claim 14, wherein said classfile is a Java compatible classfile and said adding of said third object corresponds to the addition of an invokevirtual instruction.

17. (Previously Presented) The classfile modification method of claim 1, further comprising:

adding said third object to said collection of objects at a position that corresponds to a region of said method's instructions that is executed just after said method's entry point is reached;

adding a fifth object to said collection of objects at a position that corresponds to a region of said method's instructions that is executed if an exit point of said method will inevitably be reached; and,

adding a sixth object to said collection of objects at a position that corresponds to a region of said method's instructions that will be executed if an error arises during execution of said method.

18. (Previously Presented) The classfile modification method of claim 17, wherein said classfile is a Java compatible classfile and said third, fifth and sixth objects correspond to the addition of invokestatic instructions.

19. (Previously Presented) The classfile modification method of claim 17, wherein said classfile is a Java compatible classfile and said third, fifth and sixth objects correspond to the addition of invokevirtual instructions.

20. (Currently Amended) A machine readable medium containing instructions which when executed cause a classfile modification method to be performed, said classfile modification method comprising:

before loading and using a classfile to create an object during runtime;

converting [[a]] said classfile into a collection of objects, said collection of objects including a first object that represents a method information structure found in said classfile, said collection of objections including a second object that inherits a property that said first object has, said second object representing a bytecode instruction of a method;

adding a third object to said collection of objects, said third object inheriting the properties of said first object, said third object representing a bytecode instruction to be executed by said method that, when executed by said method, invokes a pre-existing dispatcher to identify a plug-in module for said method that said method invokes to report and/or record information about said method;

adding a fourth object to said collection of objects that represents a new method information structure for said classfile, said new method information structure containing byte code instructions for a second method that registers, with said dispatcher upon loading of said classfile, an identity of said classfile's class and respective identities of methods of said classfile, said dispatcher and plug-in module being in existence prior to said loading of said classfile; and,

converting said collection of objects including said third object into a modified version of said classfile.

21-23. (Canceled).

24. (Previously Presented) The machine readable medium of claim 20, wherein said classfile modification method further comprises adding a fifth object to said collection of objects that represents a new field information structure for said classfile.

25. (Previously Presented) The machine readable medium of claim 24, wherein said field information structure is to store a numeric identification assigned to said class by said dispatcher.

26. (Canceled).

27. (Previously Presented) The machine readable medium of claim 20, wherein said third object is added to said collection of objects in a position that corresponds to a region of said method's instructions that is executed just after said method's entry point is reached.

28. (Previously Presented) The machine readable medium of claim 27, wherein said classfile is a Java compatible classfile and said adding of said third object corresponds to the addition of an invokestatic instruction.

29. (Previously Presented) The machine readable medium of claim 27, wherein said classfile is a Java compatible classfile and said adding of said third object corresponds to the addition of an invokevirtual instruction.

30. (Previously Presented) The machine readable medium of claim 20, wherein said third object is added to said collection of objects at a position that corresponds to a region of said method's instructions that is executed if an exit point of said method will inevitably be reached.

31. (Previously Presented) The machine readable medium of claim 30, wherein said classfile is a Java compatible classfile and said adding of said third object corresponds to the addition of an invokestatic instruction.

32. (Previously Presented) The machine readable medium of claim 30, wherein said classfile is a Java compatible classfile and said adding of said third object corresponds to the addition of an invokevirtual instruction.

33. (Previously Presented) The machine readable medium of claim 20, wherein said third object is added to said collection of objects in a position that corresponds to a region of said method's instructions that will be executed if an error arises during execution of said method.

34. (Previously Presented) The machine readable medium of claim 33, wherein said classfile is a Java compatible classfile and said adding of said third object corresponds to the addition of an invokestatic instruction.

35. (Previously Presented) The machine readable medium of claim 33, wherein said classfile is a Java compatible classfile and said adding of said third object corresponds to the addition of an invokevirtual instruction.

36. (Previously Presented) The machine readable medium of claim 20, wherein said classfile modification method further comprises:

adding said third object to said collection of objects at a position that corresponds to a region of said method's instructions that is executed just after said method's entry point is reached;

adding a fifth object to said collection of objects at a position that corresponds to a region of said method's instructions that is executed if an exit point of said method will inevitably be reached; and,

adding a sixth object to said collection of objects at a position that corresponds to a region of said method's instructions that will be executed if an error arises during execution of said method.

37. (Previously Presented) The machine readable medium of claim 36, wherein said classfile is a Java compatible classfile and said third, fifth and sixth objects correspond to the addition of invokestatic instructions.

38. (Previously Presented) The machine readable medium of claim 36, wherein said classfile is a Java compatible classfile and said third, fifth and sixth objects correspond to the addition of invokevirtual instructions.

39. (Currently Amended) A computing system implemented with a machine readable medium containing instructions which when executed cause a classfile modification method to be performed, said classfile modification method comprising:

before loading and using a classfile to create an object during runtime;

converting [[a]] said classfile into a collection of objects, said collection of objects including a first object that represents a method information structure found in said classfile, said collection of objections including a second object that inherits a property that said first object has, said second object representing a bytecode instruction of a method;

adding a third object to said collection of objects, said third object inheriting the properties of said first object, said third object representing a bytecode instruction to be executed by said method that, when executed by said method, invokes a pre-existing dispatcher to identify a plug-in module for said method that said method invokes to report and/or record information about said method;

adding a fourth object to said collection of objects that represents a new method information structure for said classfile, said new method information structure containing byte code instructions for a second method that registers, with said dispatcher upon loading of said classfile, an identity of said classfile's class and

respective identities of methods of said classfile, said dispatcher and plug-in module being in existence prior to said loading of said classfile; and,

converting said collection of objects including said third object into a modified version of said classfile.

40-42. (Canceled).

43. (Previously Presented) The machine readable medium of claim 42, wherein said classfile modification method further comprises adding a fifth object to said collection of objects that represents a new field information structure for said classfile.

44. (Previously Presented) The machine readable medium of claim 43, wherein said field information structure is to store a numeric identification assigned to said class by said dispatcher.

45. (Canceled).

46. (Previously Presented) The machine readable medium of claim 39, wherein said third object is added to said collection of objects in a position that corresponds to a region of said method's instructions that is executed just after said method's entry point is reached.

47. (Previously Presented) The machine readable medium of claim 46, wherein said classfile is a Java compatible classfile and said adding of said third object corresponds to the addition of an invokestatic instruction.

48. (Previously Presented) The machine readable medium of claim 46, wherein

said classfile is a Java compatible classfile and said adding of said third object corresponds to the addition of an invokevirtual instruction.

49. (Original) The machine readable medium of claim 39 wherein said adding at least one additional other object further comprises adding an additional other object at a position that corresponds to a region of said unique method's instructions that is executed if an exit point of said unique method will inevitably be reached.

50. (Previously Presented) The machine readable medium of claim 49, wherein said third object is added to said collection of objects at a position that corresponds to a region of said method's instructions that is executed if an exit point of said method will inevitably be reached.

51. (Previously Presented) The machine readable medium of claim 49, wherein said classfile is a Java compatible classfile and said adding of said third object corresponds to the addition of an invokestatic instruction.

52. (Previously Presented) The machine readable medium of claim 39, wherein said third object is added to said collection of objects in a position that corresponds to a region of said method's instructions that will be executed if an error arises during execution of said method.

53. (Previously Presented) The machine readable medium of claim 52, wherein said classfile is a Java compatible classfile and said adding of said third object corresponds to the addition of an invokestatic instruction.

54. (Previously Presented) The machine readable medium of claim 52, wherein said classfile is a Java compatible classfile and said adding of said third object

corresponds to the addition of an invokevirtual instruction.

55. (Previously Presented) The machine readable medium of claim 39, wherein said classfile modification method further comprises:

adding said third object to said collection of objects at a position that corresponds to a region of said method's instructions that is executed just after said method's entry point is reached;

adding a fifth object to said collection of objects at a position that corresponds to a region of said method's instructions that is executed if an exit point of said unique method will inevitably be reached; and,

adding a sixth object to said collection of objects at a position that corresponds to a region of said method's instructions that will be executed if an error arises during execution of said method.

56. (Previously Presented) The machine readable medium of claim 55, wherein said classfile is a Java compatible classfile and said third, fifth and sixth objects correspond to the addition of invokestatic instructions.

57. (Previously Presented) The machine readable medium of claim 55, wherein said classfile is a Java compatible classfile and said third, fifth and sixth objects correspond to the addition of invokevirtual instructions.